

# 1 Introducción

Para ofrecer a nuestros clientes la mejor experiencia de usuario, solo permitimos que puedan buscar en fechas en las que sepamos que realmente hay salidas. Al desplegarse el calendario del buscador, las fechas sin salidas aparecen deshabilitadas, y el usuario no puede seleccionarlás.



Figure 1: Calendario

Las compañías navieras ofrecen normalmente información sobre todas las salidas que hay en un mismo día, pero no cuáles días hay salidas. Por esto, para saber cuáles días podemos mostrar como disponibles, tenemos que hacer una petición por cada día y comprobar si hay salidas ese día. En el caso de la imagen mostrada, como hay 61 días mostrados en el calendario, tendríamos que hacer 61 peticiones.

Obviamente, es inviable realizar 61 peticiones por cada cliente cada vez que se abra el calendario. Para solucionarlo, en Clickferry realizamos todas las peticiones una sola vez al día y guardamos qué días hay salidas disponibles en una cache (en nuestro caso, utilizamos Redis). Cada vez que un usuario abre el calendario, esta consulta se satisface desde nuestra cache sin tener que hacer una petición a la naviera.

## 2 Recursos

Puedes utilizar la API de tadpole (<https://tadpole.clickferry.app>) para recibir las salidas de cada día. Esta API solo ofrece dos rutas:

- **ALGECEUT:** Algeciras - Ceuta
- **CEUTALGE:** Ceuta - Algeciras

Puedes solicitar las salidas para una ruta y día específicos con una petición GET a:

`https://tadpole.clickferry.app/departures?route=RUTA&time=FECHA`

Donde RUTA es uno de los códigos mencionados, y FECHA es una fecha en formato. Por ejemplo:

```
https://tadpole.clickferry.app/departures?route=ALGECEUT&time=2023-10-27
```

Devolvería una respuesta como:

```
[
  {
    "arrival": "2023-10-27T03:00:00Z",
    "operator": "BALEARIA",
    "ship": "Avemar Dos",
    "time": "2023-10-27T02:00:00Z"
  },
  {
    "arrival": "2023-10-27T10:00:00Z",
    "operator": "BALEARIA",
    "ship": "Avemar Dos",
    "time": "2023-10-27T09:00:00Z"
  },
  {
    "arrival": "2023-10-27T13:00:00Z",
    "operator": "BALEARIA",
    "ship": "Avemar Dos",
    "time": "2023-10-27T12:00:00Z"
  },
  {
    "arrival": "2023-10-27T16:00:00Z",
    "operator": "FRS",
    "ship": "Levante Jet",
    "time": "2023-10-27T15:00:00Z"
  },
  {
    "arrival": "2023-10-27T20:00:00Z",
    "operator": "ARMAS",
    "ship": "Ciudad De Ceuta",
    "time": "2023-10-27T19:00:00Z"
  }
]
```

### 3 Ejercicio: Backend

Desarrolla una aplicación backend que realice las peticiones necesarias para saber cuáles días hay salidas disponibles en los próximos 2 meses. Estas peticiones solo deberían tener que hacerse el mínimo número de veces posibles (por ejemplo, 1 vez al día, o 1 vez por cada despliegue de la aplicación).

Esta aplicación debería ofrecer un endpoint que devuelva los días en los que hay salidas disponibles de forma unificada en una sola respuesta.

## 4 Ejercicio: Frontend

Consume la API desarrollada en el backend desde una aplicación frontend para mostrar los días de los próximos 2 meses, estilando los días sin salidas de forma diferente para mostrar que están deshabilitados.

## 5 Objetivos

El principal objetivo es valorar la capacidad de manejar la interacción entre los servicios: consumir la API, guardarlos en un medio persistente y devolverlos para que sean consumidos desde el frontend. Es preferible la entrega del proyecto a través de un repositorio de GitHub.

Específicamente, no pedimos (pero podemos valorar):

- Uso de estilos en el frontend.
- Uso de alguna tecnología o lenguaje específico.
- Despliegue en producción del proyecto. Sería suficiente con instrucciones para ejecutarlo en local.